Análise de Algoritmos de Machine Learning usando Análise de Sentimentos com Cross-Domain

Emília G. Oliveira[1], Leonardo A. Monte[1]

DC – Departamento de Computação[1]

Universidade Federal Rural de Pernambuco (UFRPE) – Recife, PE – Brasil

{emilia.galdino@ufrpe.br,leonardo.monte@ufrpe.br

Abstract. *The large amount of data present on the internet, including many users opinions about products and services has created the necessity to learn with the sentiments of those consumers. As a manner to identify the users intentions within a certain context the sentiment analysis is able to extract great value informations for diverse applications. In this work we have used different domain Amazon databases, which contain consumers reviews about products sold on the webstore. We used the technique of cross domain sentiment analysis testing different machine learning algorithms with the goal of verifying the performance of the algorithms using the database after the processing proposed by this work.*

Resumo. *A grande quantidade de dados presentes na internet, sobretudo se tratando de opiniões de usuários a respeito de produtos e serviços criou a necessidade de compreender os sentimentos por trás destes consumidores. Como uma forma de detectar as impressões do usuário em um determinado texto, a análise de sentimento é capaz de extrair informações de grande valor para as mais diversas aplicações. Neste trabalho, as bases de dados de avaliações presentes em diferentes domínios do site da Amazon foram processadas e utilizadas na tarefa de análise de sentimento, por meio da técnica de cruzamento de domínios. Após o processamento dos dados e análise de sentimentos, foram aplicados diferentes algoritmos de aprendizagem de máquina, com o objetivo de verificar o desempenho dos algoritmos com o uso da base de dados após o processamento proposto.*

Introdução

A análise de sentimentos é uma área de pesquisa presente na mineração de texto que visa compreender os sentimentos presentes em textos de forma automatizada. Esta tarefa vem ganhando cada vez mais relevância devido à grande quantidade de dados disponíveis contendo avaliações de usuários sobre produtos e serviços assim como comentários sobre os mesmos [Pang and Lee, 2008]. Visando transformar essa quantidade massiva de dados, disponíveis sobretudo na internet, em conhecimento relevante para as mais diferentes aplicações, a análise de sentimentos se mostra uma tarefa que tem grande potencial de reconhecer padrões a partir de avaliações existentes e assim traçar se o comentário feito sobre determinado produto é negativo ou positivo, por exemplo.

Porém, um problema que surge em decorrência destas informações é que nem sempre existem informações suficientes para que um modelo de classificação possa ser testado de forma satisfatória e ainda restem dados para a etapa de testes [Pan et al., 2010]. Neste caso, a técnica de cruzamento de domínios (*cross domain*) é uma alternativa para esse problema, visto que é possível utilizar uma base de dados de um domínio para o treinamento do classificador e uma base de dados de outro domínio para a etapa de testes. Neste trabalho as técnicas de análise de sentimento, realizadas fazendo o uso de domínios cruzados, foram testadas usando os seguintes algoritmos de aprendizagem de máquina: Naïve Bayes, Regressão Logística, Árvore de Decisão, Random Forest, SVM (Máquina de Vetores de Suporte) e MLP (Perceptron Multicamadas). Durante a execução dos experimentos com os algoritmos de aprendizagem de máquina, foram realizadas as métricas de avaliação para que o desempenho dos algoritmos usando o processamento proposto neste trabalho pudesse ser verificado.

Quando um texto é escrito, a intenção da pessoa que o escreveu pode ser denotada a partir de diferentes aspectos do texto. Se uma pessoa assistiu a um filme e escreveu uma resenha sobre ele, a partir dos adjetivos usados, por exemplo, é possível identificar se a impressão daquela pessoa sobre o filme é positiva ou negativa. Como humanos, a tarefa de identificar a intenção e o sentimento das pessoas em um texto é algo extremamente comum e que é feito de forma cotidiana, por isso existe uma certa facilidade em identificar os sentimentos presentes na linguagem utilizada pelas pessoas.

Porém, tendo em vista a enorme quantidade de dados sendo produzidos a todo momento na internet, é inviável realizar a tarefa de análise de sentimentos manualmente, de forma que é necessário utilizar a análise de sentimentos de forma automatizada. Tendo em vista realizar a análise de textos de forma automática, existem diversos desafios no sentido de conseguir adequar a linguagem natural, que é falada por nós humanos, para um tipo de representação que seja compreensível para as máquinas. Sendo assim, no sentido de identificar de forma automática o sentimento presente em um texto, são necessárias diferentes modificações para que esta tarefa possa ser realizada. Primeiramente, em um texto existem muito além das palavras que o formam, existem números, símbolos de pontuação que apesar de fazerem sentido no contexto de comunicação entre seres humanos, não são de interesse para determinar o sentimento do texto. O principal desafio da área de análise de sentimentos e de mineração de texto é de processar as informações contidas nos textos de forma que o processo que o computador leva para compreender o texto possa ser facilitado.

1. Trabalhos Relacionados

Dentro da área de análise de sentimentos é possível encontrar diversas abordagens que são utilizadas dentro deste contexto. No trabalho de [BESBES 2018] é realizada uma análise das principais técnicas utilizadas nesta área. A área de domínio cruzado possui como características fazer o uso de diferentes bases de dados nas etapas de treino e teste dos algoritmos, e esta técnica pode ser muito útil quando no treinamento de um modelo é usada uma base de dados de um domínio e não existem dados suficientes para que possam ser

utilizados na etapa de teste, de forma que o cruzamento de domínios pode ser realizado para resolver este problema [BESBES 2018]. Para que esta técnica possa ser feita, existem diferentes formas de aplicar o domínio cruzado, dentre elas está a abordagem de encontrar pivôs, que são palavras que descrevem o mesmo sentimento e que são comuns aos domínios utilizados e que foi uma técnica utilizada nos trabalhos de [LI 2017] que também utilizou para os seus experimentos a base de dados da Amazon que foi utilizada neste trabalho. Neste trabalho, foi utilizado o conjunto de palavras comuns aos domínios utilizados e cada uma das bases de dados presente neste trabalho foi utilizada como conjunto de teste uma vez e as outras bases de dados foram utilizadas no conjunto de treinamento dos algoritmos utilizados nos experimentos.

2. Metodologia

Para a realização dos experimentos descritos neste trabalho, foram usadas quatro bases de dados de domínios diferentes provenientes das informações de avaliações de usuários disponíveis no site da Amazon. As bases de dados utilizadas neste trabalho foram as seguintes:

1. Livros;
2. Eletrônicos;
3. Cozinha;
4. DVD.

Cada uma das bases de dados é constituída de avaliações na forma de textos feitos por usuários do site da Amazon sobre diferentes produtos das seções presentes no site listadas acima e também para cada uma das avaliações, existe uma variável de classe que define se a avaliação é positiva ou negativa.

O processamento dos dados foi realizado utilizando as seguintes etapas:

1. Tokenização;
2. Remoção de stop words;
3. Palavras negativas;
4. Lematização;
5. Filtragem por classe gramatical (POS filter);
6. N-gramas.

A primeira das etapas do processamento dos dados é a tokenização, que consiste em separar os elementos presentes no texto em partes unitárias de forma que cada palavra ou símbolo de pontuação seja codificado na forma de um único token cada um. Este processo permite dividir toda a base de dados em partes menores de forma que os processamentos feitos a seguir podem ser realizados com mais facilidade. É comum encontrar em textos palavras cujo significado não possui valor dentro do contexto, mas são utilizadas como formas de ligação entre palavras mais relevantes. Estas palavras como conjunções, e até mesmo

pontuações são chamadas de *stop words* que são palavras que não possuem grande significado dentro de um texto e que exercem uma função meramente sintática dentro de uma frase, ou que não carregam muito significado dentro do texto.

A remoção das *stopwords* é feita após o processo de tokenização, justamente por ser mais fácil de encontrar essas palavras dentro de tokens. Após a identificação das *stop words* são removidos outros componentes do texto que não agregam informações úteis, como pontuações e símbolos que podem estar presentes nos textos. A etapa seguinte é a etapa de lematização que consiste em transformar as palavras para as suas formas primitivas, fazendo com que palavras similares e que seriam consideradas diferentes apesar de carregarem valor de sentimento similar, após serem transformadas para a forma primitiva serão iguais. A etapa de identificação de palavras negativas tem como objetivo fazer com que as palavras que representam negações possam ser identificadas e associadas às palavras a que fazem referência. O processo de análise das palavras de negação é importante pois, estas palavras de negação fazem referência a uma outra palavra ou sentença presente no texto, e é necessário identificar os pares de palavras negativas para que o sentimento agregado ao texto possa ser extraído de forma coerente.

Após o processo de identificação de palavras negativas é feita a filtragem por classe gramatical (POS filter) que busca compreender quais são as classes gramaticais de cada uma das palavras presentes na base de dados A aplicação da filtragem por classe gramatical visa melhorar os resultados de análise de sentimento, preservando apenas as classes gramaticais que agregam maior valor à análise de sentimentos, sendo elas: verbos, advérbios, substantivos e adjetivos. A última etapa no pipeline de pré-processamento das bases de dados é a criação dos n-gramas, que são concatenações realizadas entre as palavras, que estão na forma de tokens, de forma a agregar maior valor de sentido ao unir em grupos de n palavras as palavras que aparecem em sequência no texto.

3. Experimentos

Os testes foram realizados em diferentes algoritmos de agrupamento sendo eles: Naïve Bayes, SVM(Máquina de Vetores de Suporte) com função de kernel RBF, Árvore de Decisão, Random Forest e Regressão Logística. Os experimentos foram realizados de forma a seguir as seguintes etapas:

1. Pré-processamento da base de dados;
2. Estratificação da base de dados;
3. Execução dos algoritmo de aprendizagem de máquina;
4. Geração dos gráficos com as análises experimentais.

Primeiramente os dados foram processados de forma a extrair de forma mais eficiente os sentimentos presentes no texto e após esse processamento as bases de dados foram estratificadas usando um total de 1800 amostras de cada uma das bases de dados para serem usadas nos experimentos. Para a etapa de execução dos algoritmos, foram feitos testes usando as seguintes configurações de distribuição das bases de dados:

1. Bases de dados de treino: Livros, Cozinha e DVD, base de dados de teste: Eletrônicos;
2. Bases de dados de treino: Livros, Cozinha e Eletrônicos, base de dados de teste: DVD;
3. Bases de dados de treino: Cozinha, Eletrônicos, Livros, base de dados de teste: Livros;
4. Bases de dados de treino: DVD, Eletrônicos, Livros, base de dados de teste: Cozinha.

Para cada um dos conjuntos de dados foram realizados a divisão das amostras para treino e teste usando o k-fold com 15 iterações sendo executado em um 10-fold de forma que foram feitos um total de 150 experimentos. Os experimentos foram realizados tanto com algoritmos com processos estocásticos e não-estocásticos, de forma que para os algoritmos que possuem processos não-estocásticos por não apresentarem variações nos seus resultados de acordo com as diferentes amostras utilizadas não foram gerados os boxplots com as métricas de análise, de forma que essa análise foi feita apenas para os seguintes algoritmos: Árvore de decisão, Random Forest e MLP com backpropagation.

O algoritmo de **MLP** consiste em uma rede neural, que é um tipo de algoritmo cujo funcionamento é inspirado no comportamento das sinapses realizadas pelo cérebro. Nos algoritmos de redes neurais, existem entradas que são os dados fornecidos pela base de dados e que são codificados de acordo com funções e pesos que são codificados dentro da rede neural e que após a realização destes cálculos é passada uma função de limiar que irá decidir qual será o resultado a ser retornado como saída.

O algoritmo de **Árvore de decisão** é baseado na estrutura de dados de uma árvore e tem como principal objetivo criar uma configuração de árvore, na qual os nós são as características e as arestas são os valores, que seja capaz de classificar da melhor forma possível os dados. Para este processo são feitos cálculos relacionados às características e de acordo com o grau de eficiência na divisão dos dados em classes distintas, são selecionadas as melhores características para construção da árvore que irá realizar a tarefa de classificação.

O algoritmo de **Random forest** possui funcionamento inspirado no algoritmo de Árvore de Decisão, com a diferença que na Random Forest são feitas várias árvores de decisão de forma a encontrar a melhor forma de classificação dos dados.

O algoritmo **Naïve Bayes** é um classificador probabilístico baseado no Teorema de Bayes que assume que não existe dependência entre as features do modelo.

O algoritmo **Máquina de Vetores de Suporte** é um classificador que busca criar um hiperplano que divida os dados da melhor maneira possível. O hiperplano busca maximizar a sua distância em relação aos elementos mais próximos de cada classe.

O algoritmo de **Regressão Logística** é uma técnica estatística que tem como objetivo produzir, a partir de um conjunto de observações, um modelo que permita a predição de valores tomados por uma variável categórica, frequentemente binária, a partir de uma série

de variáveis explicativas contínuas e/ou binárias.

Para cada um dos algoritmos foram realizadas as seguintes métricas de avaliação:

1. Acurácia;
2. F-score;
3. Precision;
4. Recall;
5. Tempo de execução.

Para cada uma destas métricas foram calculados os seus respectivos valores em cada uma das execuções dos experimentos para os algoritmos testados neste trabalho. Após a execução de todos os experimentos, foram gerados os gráficos dos boxplots dos valores de cada uma das métricas por conjunto de teste. Também foi realizado um teste estatístico com o objetivo de verificar a hipótese de que os resultados apresentados pelos algoritmos foram realizados por características dos próprios algoritmos e não de forma aleatória. O teste estatístico realizado foi o teste de Friedman e os resultados dos testes de acordo com cada configuração de bases de dados são mostrados na seção a seguir.

4. Resultados experimentais

Após a realização dos experimentos, foram gerados gráficos no modelo de gráficos de *boxplot* para mostrar os resultados obtidos de acordo com cada uma das métricas utilizadas nos experimentos realizados.

4.1 Livros como base de dados de teste

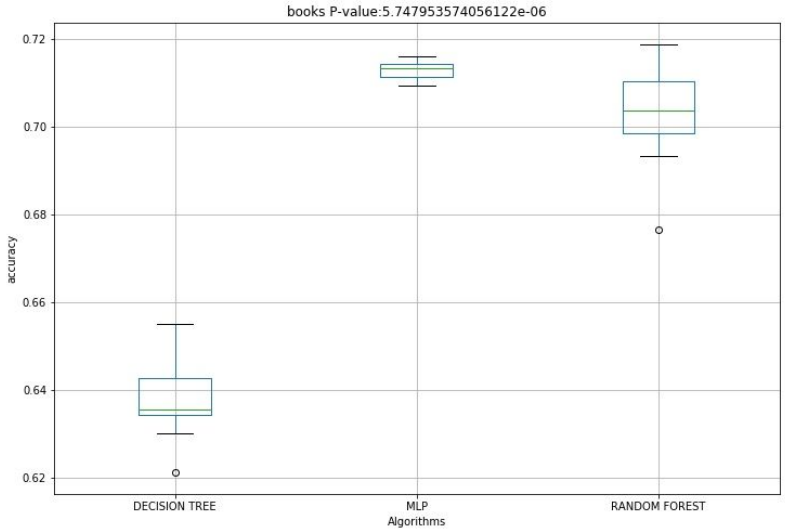

Figura 1. Boxplot das acurácias obtidas pelos algoritmos com processos estocásticos com a base de Livros como teste.

Algorithm	Accuracy	Precision	F1	Recall	Time
DecisionTree	63,75	63,32	64,35	65,44	8,814
MLP	71,3	70,65	71,75	72,89	41,943
NaiveBayes	62,89	62,42	63,58	64,78	0,75
RandomForest	70,39	72,83	68,71	65,07	1,574
RegLog	74,72	74,53	74,82	75,11	2,311
SVM	74,33	72,3	75,45	78,89	178,023

Tabela 1. Tabela de resultados médios obtidos pelos algoritmos com a base de Livros como teste.

4.1. Eletrônicos como base de dados de teste

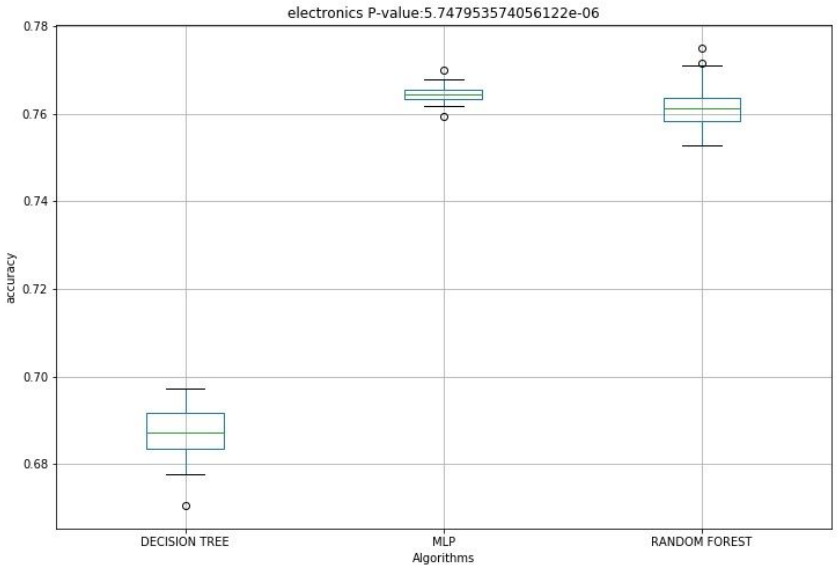

Figura 2. Boxplot com os resultados de acurácia obtidos pelos algoritmos com processos estocásticos usando a base de Eletrônicos como teste.

Algorithm	Accuracy	Precision	F1	Recall	Time
MLP	76,44	75,57	76,84	78,15	24,511
DecisionTree	68,64	68,81	68,51	68,22	3,569
NaiveBayes	70,17	67,95	71,9	76,33	0,489
RandomForest	76,2	82,12	73,78	67,02	0,941
LogReg	79,61	80,25	79,39	78,56	1,494
SVM	82,39	84,66	81,79	79,11	160,27

Tabela 2. Tabela de resultados médios obtidos pelos algoritmos usando a base de Eletrônicos como teste.

4.2. Cozinha como base de dados de teste

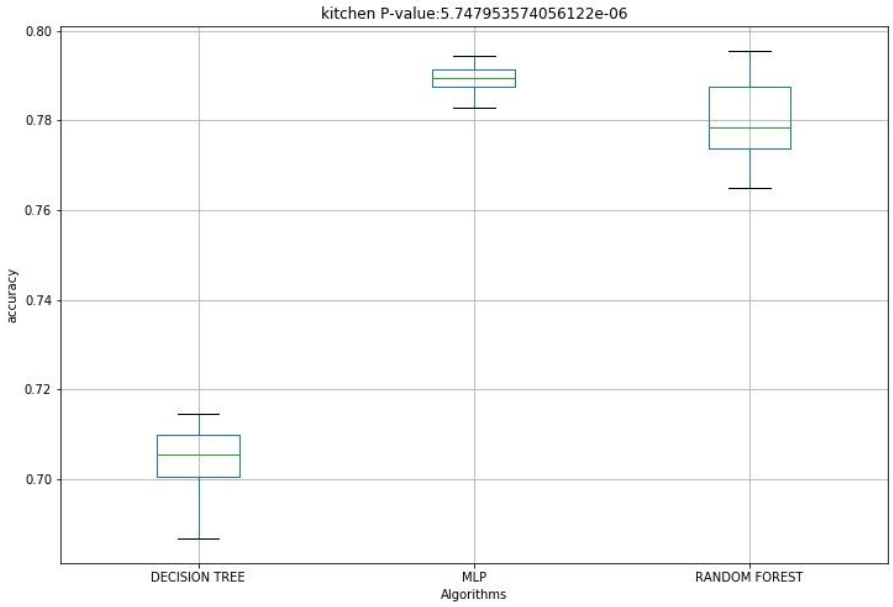

Figura 3. Boxplot com os resultados de acurácia obtidos pelos algoritmos com processos estocásticos usando a base de Cozinha como teste.

Algorithm	Accuracy	Precision	F1	Recall	Time
MLP	78,93	76,72	79,76	83,07	29,1
DecisionTree	70,46	68,82	71,68	74,81	3,546
NaiveBayes	68,33	64,81	71,7	80,22	0,394
RandomForest	78,01	78,61	77,79	77,01	0,875
LogReg	81,56	78,74	82,42	86,44	1,436
SVM	82,28	79,92	82,95	86,22	110,726

Tabela 3. Tabela de resultados médios obtidos pelos algoritmos usando a base de Cozinha como teste.

4.3. DVD como base de dados de teste

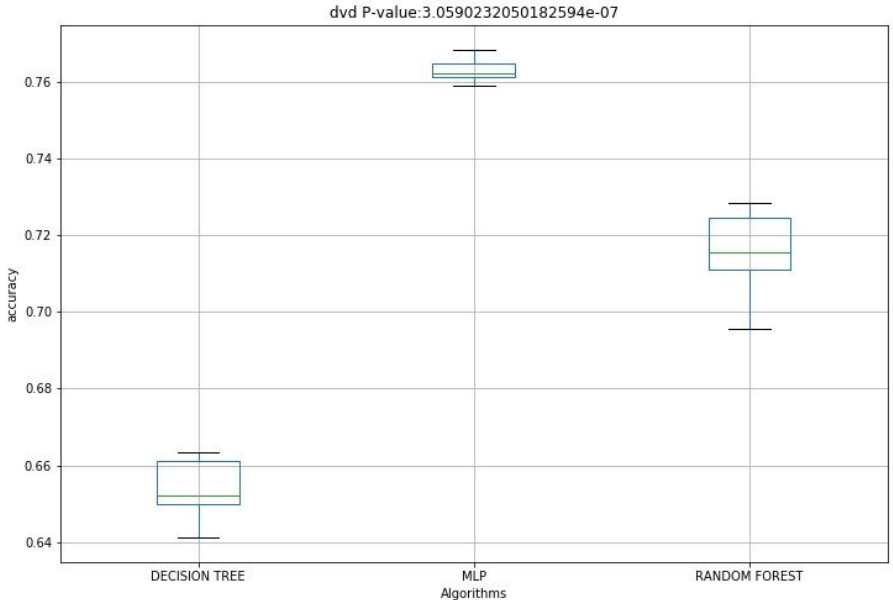

Figura 4. Boxplot com os resultados de acurácia obtidos pelos algoritmos com processo estocástico usando a base de DVD como teste.

Algorithm	Accuracy	Precision	F1	Recall	Time
MLP	76,29	73,81	77,46	81,5	43,943
DecisionTree	65,35	64,83	65,95	67,12	9,002
NaiveBayes	67,78	65,56	69,92	74,89	0,68

RandomForest	71,51	74,49	69,67	65,47	1,567
LogReg	79	76,8	79,83	83,11	2,217
SVM	78,78	77,73	79,17	80,67	184,624

Tabela 4. Tabela de resultados médios obtidos pelos algoritmos usando a base de DVD como teste.

5. Conclusão

Após a análise empírica dos dados presentes nos gráficos gerados a partir dos experimentos, e também da análise estatística utilizando o teste de Friedman a conclusão é de que dos algoritmos testados neste trabalho, o algoritmo que obteve melhores resultados foi a Máquina de vetores de suporte. Porém, em comparação com o algoritmo de Regressão Logística, que também obteve resultados satisfatórios, o tempo de execução da Máquina de vetores de suporte é muito maior, de forma que caso fossem utilizadas mais instâncias da base de dados, ou fossem realizados mais testes, o tempo necessário para concluir tais experimentos seria muito custoso. De todos os algoritmos testados no experimento, os que obtiveram os piores resultados foram os algoritmos de Árvore de Decisão e Naive Bayes, ficando quase 18 pontos abaixo da média de outros algoritmos com melhores resultados.

6. Referências

PANG, B., Lee, L. (2008) Opinion mining and sentiment analysis. Foundations and Trends R in Information Retrieval, 2(1–2):1–135.

LIU, B. (2012) Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies, 5(1):1–167.

PAN, S., NI, X., SUN, J., YANG, Q., CHEN, Z. (2010) Cross-domain sentiment classification via spectral feature alignment. In Proceedings of the 19th International Conference on World Wide Web, pages 751–760. ACM.

BESBES, A. (2018). Overview and comparison of traditional and deep learning models in text classification.

LI, Z., ZHANG, Y., WEI, Y., WU, Y., YANG, Q. (2017) End-to-End Adversarial Memory Network for Cross-domain Sentiment Classification.